Planned NIST Support for the I++ DME Interface Spec

July 2-3, 2002
Frankfurt, Germany

John Horst
National Institute of Standards and Technology (USA)

Phases of NIST support

- Phase 1: build and supply tools to support initial implementation development, review versions of the spec, interface with related standards efforts, and facilitate interaction
- Phase 2: define and maintain a test suite for use with a Distributed Testbed, review versions of the spec, interface with related standards efforts, and facilitate interaction

Phase 1 support

- Tools to facilitate server implementation development
  - a “client-side utility”
  - a set of command files to use in testing
  - metrics and analysis tools
  - procedures

Phase 1 support

- Tools to facilitate client implementation development
  - a “server-side utility”
  - measurement “programs” (pseudo-code)
  - results logging software (for inclusion in implementation software)
  - metrics and analysis tools
  - procedures

Phase 1 support

- Tools to facilitate both server and client utilities
  - artifact(s)
  - a specification of command file syntax

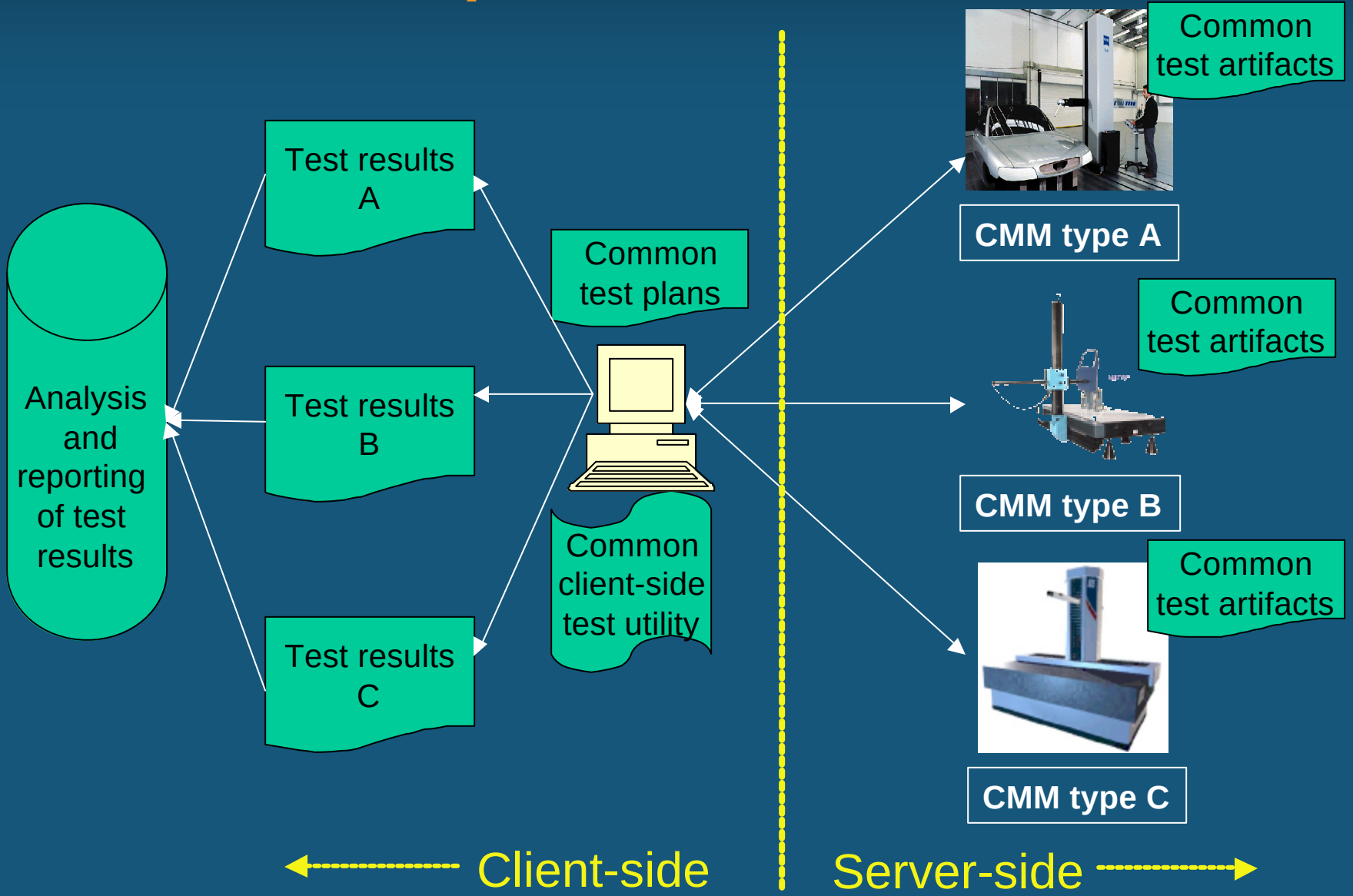
Phase 2 support

- With I++ team, define how phase 1 tools can be used to support tests
- Create a test suite consisting of
  - Functionality, conformance, and interoperability tests
  - Test cases (inspection plans and artifacts)
  - Common test software utilities
  - Analysis tools and metrics
  - Testing and validation procedures and schedules
- NIST will NOT provide a testing service
- Test suite iterative with specification

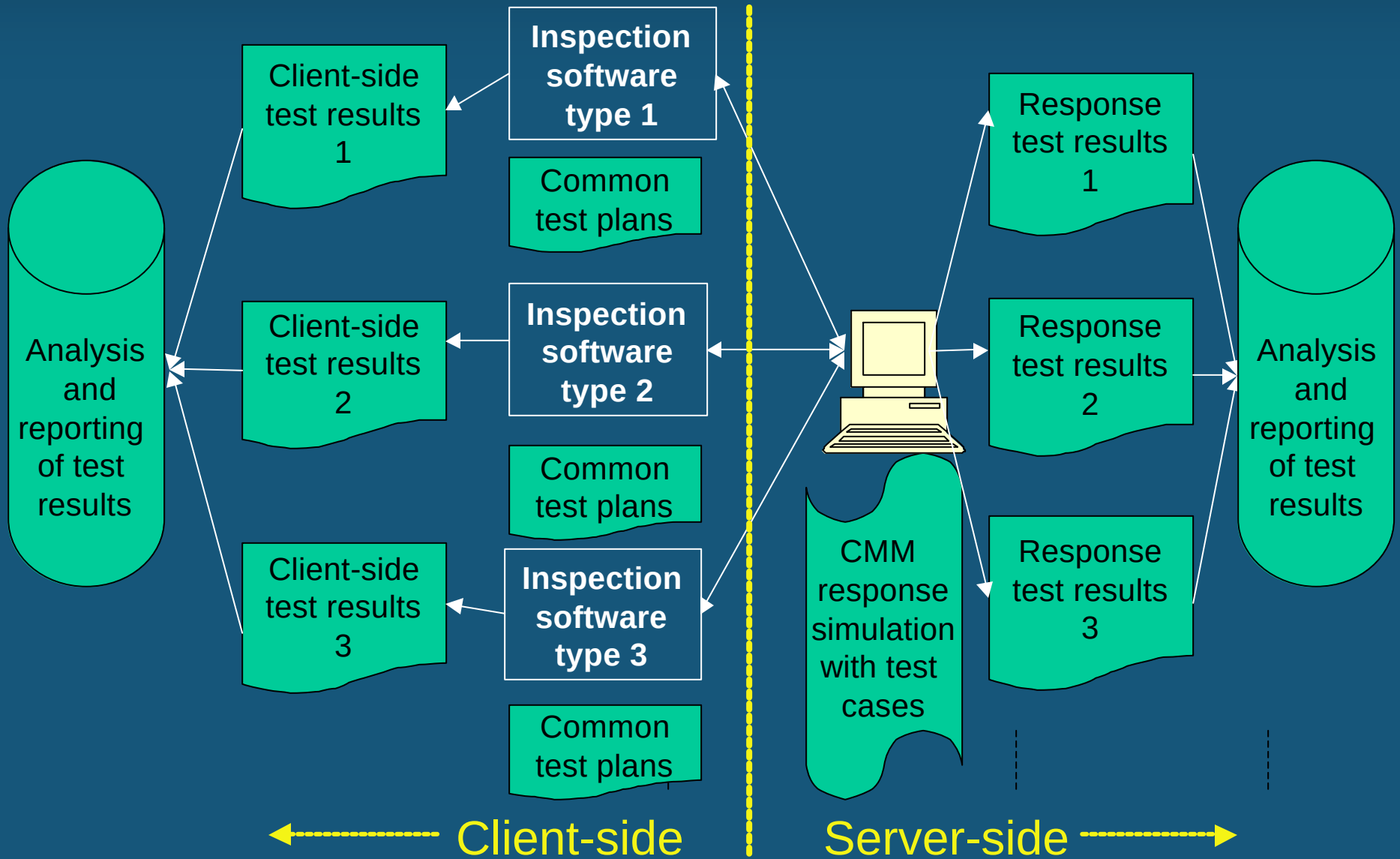
Why Test Suite?

- Specification alone insufficient for interoperability
- Reduces variability in each test
- Allows application of quantifiable metrics
- More cost to changes after publication of the specification
- Facilitates high quality, timely standard

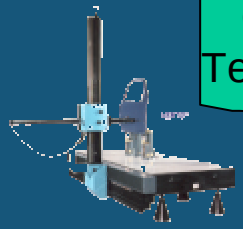
Conformance tests for server-side implementations

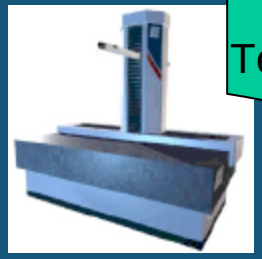


Conformance tests for client-side implementations



Interoperability Tests





Common Test Plans

Inspection Software 1

Common Test Plans

Inspection Software 2

Common Test Plans

Inspection Software 3

Common Test Artifacts

CMM type A

Common Test Artifacts

CMM type B

Common Test Artifacts

CMM type C

Test Results A1

Test Results A2

Test Results A3

Test Results B1

Test Results B2

Test Results B3

Test Results C1

Test Results C2

Test Results C3

Analysis and Reporting

Analysis/Metrics

- Reference log files
- Log file utilities for test automation
- Metrics used in automated analysis
- Some manual analysis unavoidable
- Need to define “success” as a group

Miscellaneous

- Video for test results visualization
- Command/response currently non-real-time
- Distributed testbed essential
  - Saves money, saves time, and improves quality of tests
  - Allows for latest upgrades
  - Avoids costly unnecessary duplication of equipment and software
  - Good (and cost-effective) division of human expertise and labor

Current status

- NIST client-side utility is now I++ DME spec-compliant, but untested
- NIST server-side utility not done
- Functionality test not defined
- Only a few test cases defined for client-side utility
- Test artifact complete
- Initial metrics and procedures defined for conformance tests
- NIST node of distributed testbed in progress

Suggestions for development group organization

July 2-3, 2002
Frankfurt, Germany

John Horst
National Institute of Standards and Technology (USA)

I++ DME implementation and testing team: organizational ideas

- Communicate via email, conference calls, and net meetings
- Form an executive working group of approximately 4 to 6 people selected randomly for limited term (say, 4 months)
- Decisions made by majority vote of standing executive committee (I++ customers' role??)
- NIST representative permanently sits on the executive committee with no vote

Demo of NIST client-side utility

July 2-3, 2002
Frankfurt, Germany

John Horst
National Institute of Standards and Technology (USA)